WWW.SCIAM-MAGAZINE.NAROD.RU

Фантастический полет над необитаемой базой данных

А. К. ДЬЮДНИ

Я ЛИХОРАДОЧНО изучал приборную панель самолета «Чессна 182». Наверное, с моей стороны было неразумно пытаться совершить полет в одиночку без надлежащей подготовки. Может быть, следовало взять с собой опытного пилота? На чикагском аэродроме Мейгс-Филд не было видно ни души, даже ни один механик не вышел покурить на солнышке. Единственно на что я мог положиться в этой сумасбродной затее — это на руководство для пилота, на изучение которого я потратил не больше 10 минут.

Думаю, что я чувствовал бы себя увереннее, если бы у меня в руках была ручка управления или штурвал. Еще ребенком я по многу раз смотрел фильмы о приключениях летчиков-асов и хорошо запомнил основные приемы управления самолетом. Однако в этом самолете все было по-другому. Он управлялся с клавиатуры стоящего передо мной терминала.

Со вздохом я раскрыл страницу 46 руководства. Клавиша Ф2 слегка приоткрывает дроссель, клавиша Ф4 широко раскрывает его. Будучи человеком не совсем безрассудным, я с опаской нажал на клавишу Ф2. Мягкий шум двигателя несколько усилился. Взлетная дорожка побежала мне навстречу, и на какое-то мгновение мною овладела паника. Самолет почти сразу же начал уходить в сторону от бетонной полосы. По невежеству я не прочел ничего о руле горизонтального управления или о переднем шасси. Спасти меня могли только элероны — клавиша 4 (левый крен) и клавиша 6 (правый крен). К моему полному изумлению, одна из них стала медленно, но верно возвращать самолет на взлетную полосу.

Всего в ста метрах впереди показались голубые воды озера Мичиган. Прибавляю газ — Ф2, Ф2, Ф2, Ф2. Прибавляю еще. Мотор звучит все мощнее, но самолет упорно отказывается взлетать. Взлетная полоса побежала подо мной еще быстрее, и меня опять охватила паника. Какая же волшебная клавиша поднимет меня в воздух? Вся моя жизнь пронеслась предо мной. Сцены из далекого детства: вот я управляю моделью самолета, это можно было делать лишь с помощью двух закрылков на хвосте. Руль высоты! Я стал лихорадочно искать «руль высоты» в своем руководстве. Этот руль управляет углом атаки самолета, при его помощи можно приподнять нос самолета. Нажимаю на клавишу 2, и как раз вовремя, еще немного и передо мной разверзлась бы водяная могила. Страшная граница между сушей и водой стала отступать. Самолет был в воздухе! Он набирал высоту как будто по своей воле. Внизу все четче проступала перспектива западной береговой линии озера Мичиган, и в течение какого-то времени я продолжал полет в северном направлении, загипнотизированный своим достижением.

Как уже, наверное, догадались многие читатели, моя кабина находилась в комнате, а роль ручек управления играли клавиши на терминале моего персонального компьютера, который выполнял программу имитации полета. Однако знакомая обстановка ничуть не снижала остроту ощущений. Тот факт, что и самолет, и ландшафт лишь имитировались компьютером, даже усиливал воображение. В этих условиях можно было совершать такие действия, на которые никто бы не отважился, сидя в кабине настоящего самолета. Ведь счет, предъявленный за аварию, будет тоже воображаемым. Я сделал вираж в юго-западном направлении и стал приближаться к Чикаго. Мне захотелось пронестись над Мичиган-авеню и произвести впечатление на многочисленных водителей и пассажиров движущегося по этой оживленной улице транспорта. Часы, вмонтированные в приборную панель самолета, показывали 4 ч 30 мин, как раз начало вечернего часа пик.

Приближаясь к Чикаго, я стал замечать что-то неладное. Были видны лишь несколько зданий, расположенных вдоль берега озера, а на улицах никакого транспорта. Людей тоже не было видно. Может быть, началась атомная война? Я заметил, что принял за часы высотомер с двумя стрелками, но даже несмотря на то, что сейчас было значительно раньше 4 ч 30 мин, полное отсутствие людей на улицах такого большого города казалось странным. Хотя и неохотно, но я пришел к выводу: в программе имитации полета просто не нашлось места для людей. Ее база данных и без того была перегружена описанием североамериканского ландшафта, не говоря уже о всевозможных географических подробностях, необходимых для моделирования четырех больших городов и более 20 аэродромов.

В последующих полетах я уже все меньше и меньше думал о недостатке в людском обществе. Слишком интересен был сам полет. В этом со мной, наверное, согласятся многие из тех, кто совершал подобные воздушные путешествия. Программа называется FLIGHT SIMULATOR (Имитатор полета), и в течение уже многих месяцев она была одной из наиболее популярных, если не самой популярной из всех занимательных программ для персональных компьютеров. Существует немало других программ имитации полета, которые можно купить или приобрести бесплатно в различных ассоциациях. Однако как FLIGHT SIMULATOR, так и ряд других аналогичных программ своей популярностью обязаны Б. Артвику, инженеру-программисту из Шампейна (шт. Иллинойс). Более чем кто-либо другой Артвик способствовал «взлету» программ имитации полета, и благодаря его усилиям они теперь могут выполняться на персональных компьютерах. В течение долгого времени компьютерная имитация полета применялась исключительно с целью обучения летчиков (см. статью Р. Хабера «Моделирование условий полета», с. 64—73). Теперь же эти средства доступны практически каждому.

Рассмотрим, в чем должны заключаться особенности таких программ, чтобы поддерживать у пользователя иллюзию реального полета. Во-первых, они должны создавать и несколько раз в секунду воссоздавать ландшафт, видимый из какой-то фиксированной точки. По мере того как моделируемый аэроплан бороздит моделируемые небеса, точка наблюдения перемещается, и поэтому черты проплывающего на экране ландшафта должны вычисляться заново. Более того, для каждой точки наблюдения возможны различные углы зрения. Угол зрения определяется результатами имитации физических особенностей аэроплана, находящегося в полете. Например, если самолет делает крен в сторону, чтобы совершить поворот, угол зрения, так же как и точка наблюдения, должен соответственно меняться. В основном цикле вычислений применяется ряд трюков и осуществляется поиск компромиссных решений, что само по се-

WWW.SCIAM-MAGAZINE.NAROD.RU

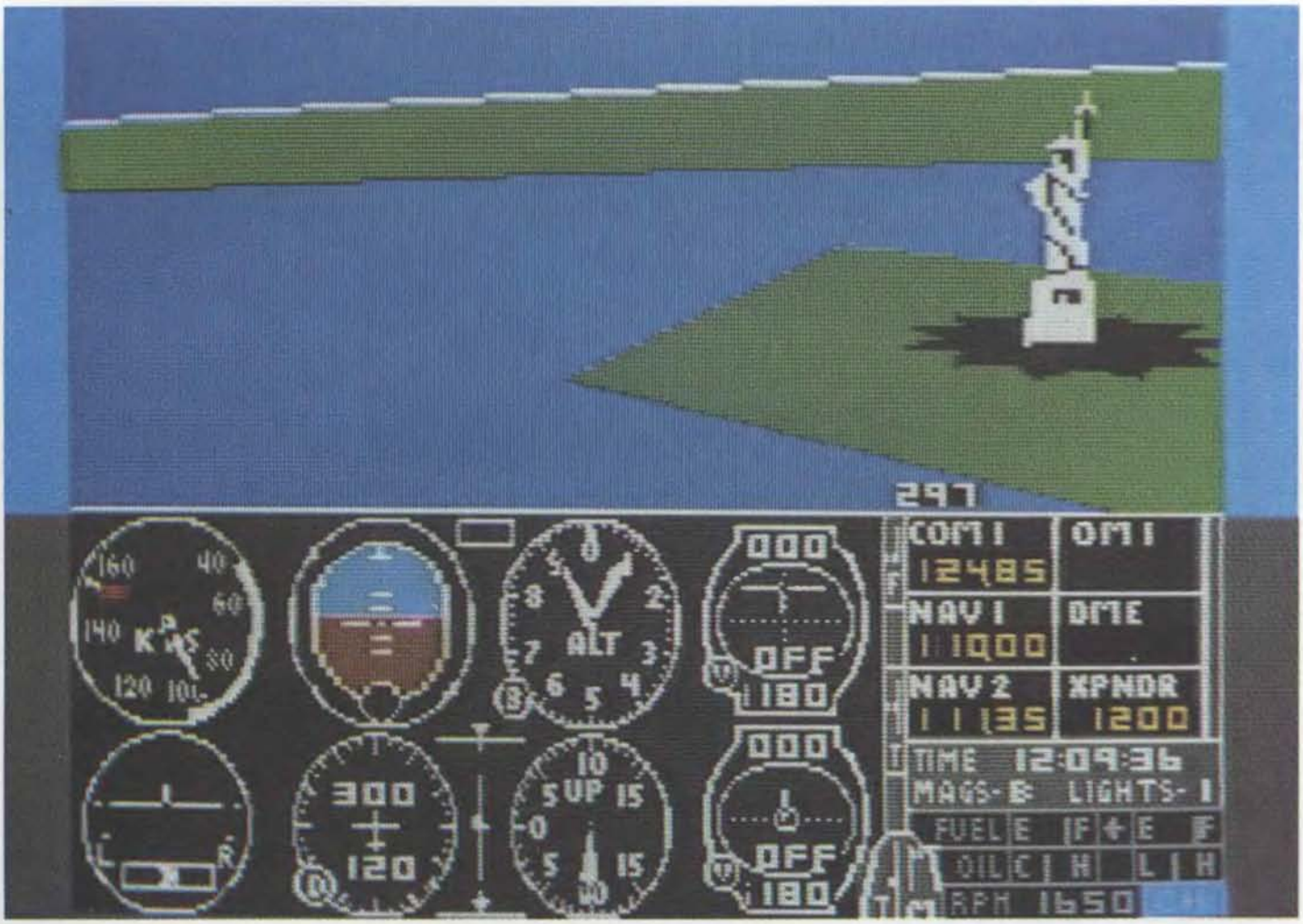
бе не менее увлекательно, чем имитация полета в целом.

Географическая база данных является одним из главных компонентов программы FLIGHT SIMULATOR. Она включает обширный список объектов, необходимых для воссоздания ландшафта территории, над которой желает полетать пользователь (или в которую он не хочет врезаться). Вы можете получить доступ к «сцене» вашего полета, обратившись к иерархической информационной системе и прослеживая за данными «сверху вниз». В массиве памяти под названием USA (США) хранятся названия районов, таких, как Чикаго, Нью-Йорк и Сиэтл. Положение самолета задается тремя координатами: x — широта, y — высота над уровнем моря (в метрах) и z — долгота. Программа FLIGHT SIMULATOR сравнивает эти координаты с предельными значениями, ограничивающими каждый район в массиве США. Если, например, x и z задают точку, принадлежащую району Чикаго, то программа выполняет операцию, называемую загрузкой базы данных. В результате этой операции список, в котором хранится информация о районе Чикаго, загружается с магнитного диска в оперативную память компьютера.

Когда в ходе своего первого полета я сделал вираж, повернув по направлению к Чикаго, сначала я не увидел никаких зданий. Однако некоторое время спустя показалась башня Сирз-Тауэр. Затем стали проступать и ее детали. В этом как раз и заключается стратегия программы, позволяющая ей уложиться в пределы имеющегося объема памяти и быстродействия; особенности ландшафта не отображаются на экране до тех пор, пока в них нет реальной необходимости. В программу встроены две специальные тестирующие процедуры, управляющие появлением и исчезновением на экране тех или иных деталей ландшафта. Первая из них *ifin2d* следит лишь за двумя координатами — широтой и долготой самолета. Выше уже говорилось, каким образом географические районы ассоциируются с этими двумя координатами.

Вторая тестирующая процедура *ifin3d* определяет расстояние до заданной детали ландшафта, и если она находится достаточно близко, программа отображает ее на экране. По мере того как я приближался к Чикаго, процедура *ifin3d* постоянно вычисляла расстояние от самолета до тех или иных зданий или других ориентиров города, видимых с воздуха. Когда мой самолет приблизился к башне Сирз-Тауэр на расстояние 12 800 м, на экране внезапно появилась миниатюрная копия этого сооружения в виде нескольких мерцающих точек. Когда же расстояние до башни уменьшилось до 7680 м, появились уже некоторые неразличимые ранее детали. Для каждых трех новых значений координат FLIGHT SIMULATOR быстро просматривает список, хранящий все видимые особенности ландшафта, и проверяет их согласно критериям отображения объектов, на которых основана процедура *ifin3d*. Если объект удовлетворяет критериям, программа отображает его на экране таким, каким он должен быть виден из данной точки. Если же объект не проходит теста, он не отображается, и программа переходит к проверке следующего объекта в списке.

Вообще говоря, каждый объект или



Вид статуи Свободы из кабины самолета, построенный программой FLIGHT SIMULATOR

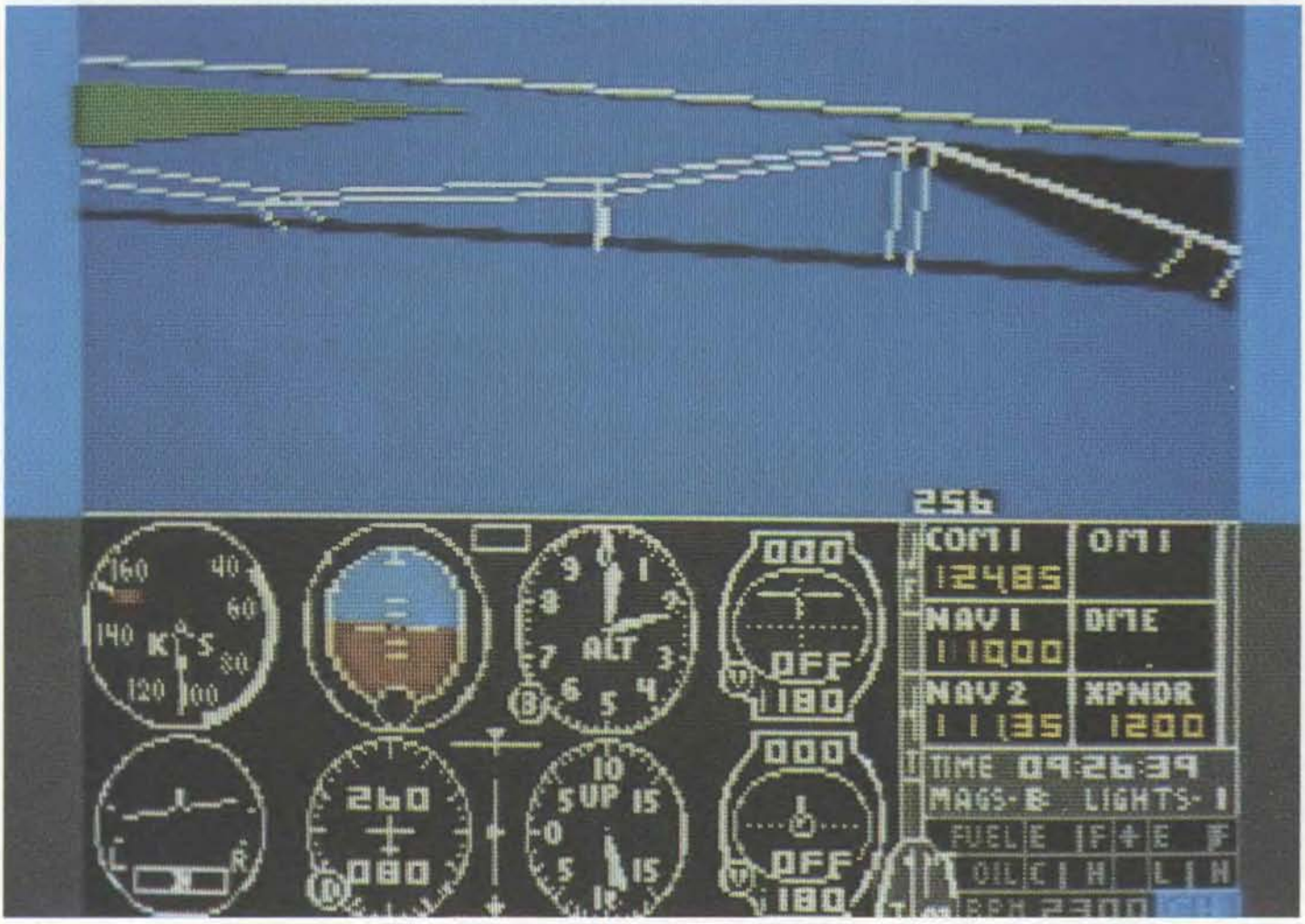
WWW.SCIAM-MAGAZINE.NAROD.RU

каждая особенность моделируемого ландшафта определяется в программе списком точек, линий и поверхностей. Поскольку изображение вычисляется пять раз в секунду, достаточно часто для того, чтобы воспользоваться инерционностью человеческого зрения, линии и поверхности должны отображаться за минимально возможное время. Поэтому даже элементарные графические операции были реализованы в программе самым тщательным образом. Артвик описал эти операции в своей замечательной книге «Microcomputer Displays, Graphics, and Animation» («Микрокомпьютерные изображения, графика и мультипликация»).

Например, в основном методе, применяемом для изображения линии, операции умножения и деления не применяются, поскольку для их выполнения требуется в 10 и даже в 100 раз больше времени по сравнению с операциями сложения и вычитания. В данном случае метод изображения линии основан на тщательном анализе временных затрат на каждую выполняемую команду. Получившийся в результате остроумный алгоритм изображения линии по существу состоит лишь из четырех команд: сложения, сравнения, перехода и команды отображения. Пусть, например, точка с координатами (a, b) была только что высвечена на линии с (положительным) наклоном 1/4. Программа входит в цикл, в котором она повторно прибавляет 1 к значению координаты a и 1/4 к специальной сумме, называемой ошибкой прослеживания, и затем тестирует текущее значение этой суммы. Если ошибка прослеживания оказывается большей или равной какому-нибудь целому числу, то программа выходит из цикла и высвечивает следующую точку, в данном случае с координатами ($a + 4$, $b + 1$).

Поверхности программа изображает посредством метода, который едва ли поддается описанию, но зато чрезвычайно быстро реализуется в компьютере. В нескольких словах этот процесс можно представить следующим образом. Программа изображает многоугольник, ограничивающий данную поверхность, и затем «заштриховывает» его внутреннюю область. После того как вычислена форма многоугольника, специальная графическая процедура вначале определяет сканирующие линии, пересекающие на экране полученный многоугольник. Для каждой такой линии сканирования определяются пересекаемые ею стороны многоугольника, а затем эти стороны сортируются (упорядочиваются) в соответствии с порядком, в котором линия сканирования пересекает их слева направо. Затем процедура рассматривает каждый пиксел, или элемент изображения, на линии сканирования по порядку слева направо. Когда линия встречается с первой стороной многоугольника, она начинает высвечивать точки определенного цвета, выбранного для представления данной поверхности. Как только линия сканирования доходит до следующей стороны, отображение точек прекращается. При встрече со следующей стороной отображение возобновляется и т.д.

Описанный метод включения-выключения отображения вдоль линии сканирования экрана основан на фундаментальном положении топологии на плоскости: любой многоугольник делит плоскость на две части — внут-



По смертельно опасному курсу на Манхаттенский мост

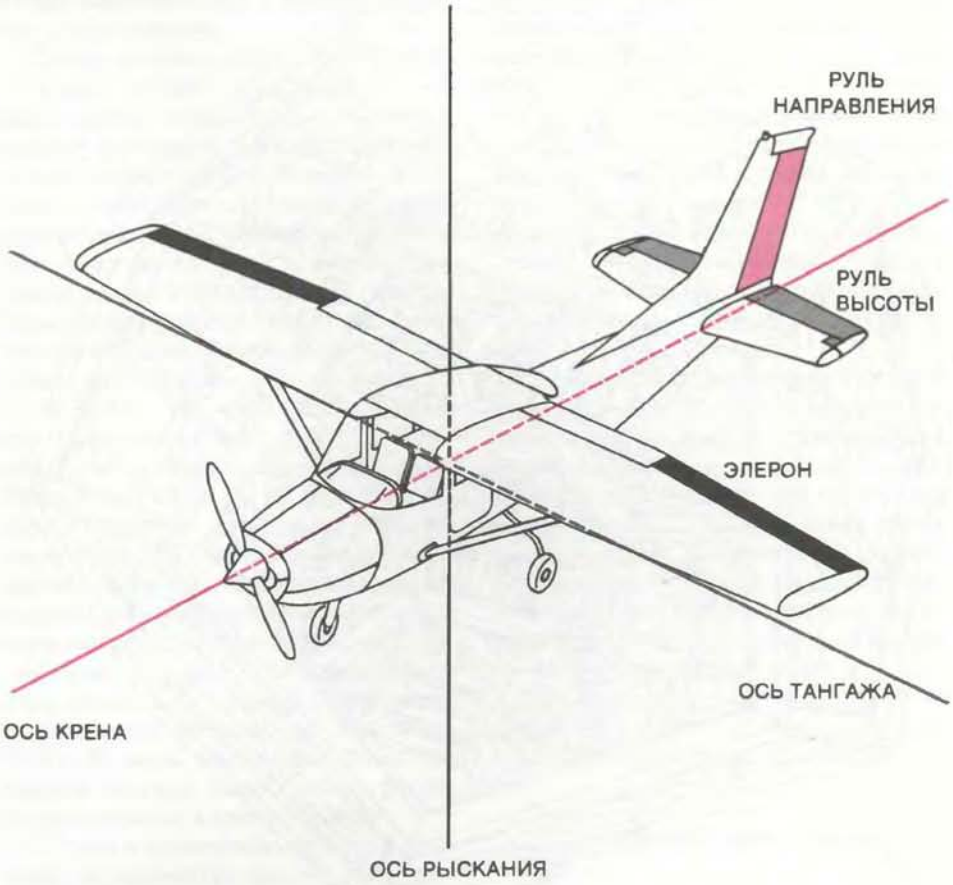
WWW.SCIAM-MAGAZINE.NAROD.RU

реннюю и внешнюю. Поэтому любая линия, пересекающая сторону многоугольника, переходит из внутренней части (принадлежащей отображаемой области) во внешнюю (не принадлежащую ей) или наоборот. Большинство поверхностей, отображаемых программой FLIGHT SIMULATOR, ограничены сравнительно простыми многоугольниками. Например, после того как я покинул зону аэродрома Мейгс-Филд, передо мной до самого горизонта протянулась береговая линия озера Мичиган. Эта линия представляла собой несколько сторон многоугольника, очерчивающих залив, ограниченный отмелью, тянущейся в северном направлении. Голубые линии воды в заливе прерывались в зоне отмели и затем снова возникали, заполняя главную часть озера.

Специалисты, работающие с графическими программами, часто размышляли над так называемой проблемой скрытой линии. Каким образом программа должна изображать трехмерное непрозрачное тело, чтобы видна была только та грань, которая находится ближе к наблюдателю? Некоторые методы, применимые к этой задаче, требуют решения трудоемких уравнений, на что уходит слишком много времени. FLIGHT SIMULATOR едва ли может себе это позволить. Решение Артвика, применимое лишь к простым ландшафтам, заключается в том, чтобы при изображении объекта ближайшая к наблюдателю грань заштриховывалась последней. Части объекта, которые должны оставаться невидимыми для наблюдателя, просто затираются теми. которые должны быть видны. Представление двух башен Международного торгового центра в Нью-Йорке дает нам хороший пример применения этого грубого, но эффективного метода. При любом угле зрения ближайшие части здания закрывают более далекие и ближняя башня «затмевает» дальнюю.

После того как я узнал, что Чикаго необитаем, я решил вернуться на Мейгс-Филд. Владелец самолета, наверное, уже беспокоится. И все же я хотел еще многому научиться. Во-первых, мне было как-то не по себе оттого, что я не знал назначения нескольких рычагов управления. Во-вторых, я так и не попытался выяснить какую-нибудь фигуру высшего пилотажа. Для этой цели я вылетел в зону озера Мичиган.

Моим излюбленным средством управления были элероны. При нажатии на клавишу левого элерона самолет делает крен влево: горизонт передо мной наклонился вправо, и весь



Три оси, вокруг которых может поворачивать самолет

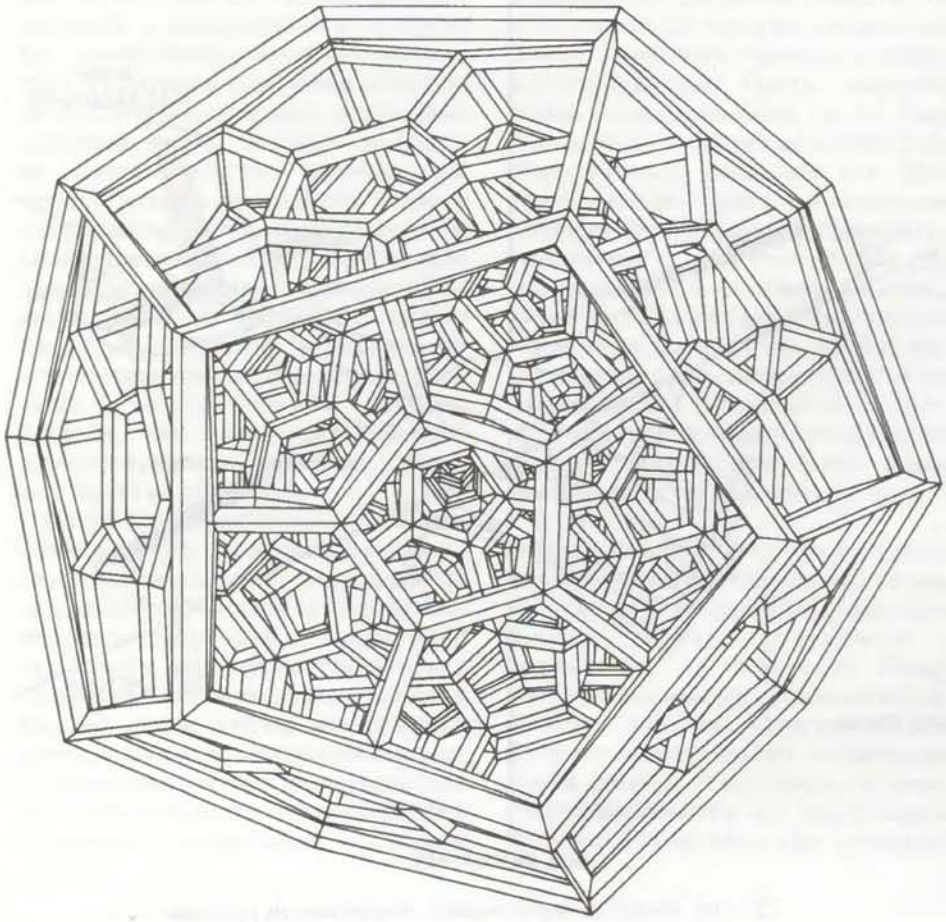
ландшафт, казалось, начал сползать по склону. Не противореча законам аэродинамики, нос «Чессны» все это время поворачивал влево. Вернув крыло в горизонтальное положение и продолжая полет по прямой, я нажал несколько раз на клавишу 8 и почти физически почувствовал, как нос самолета нырнул вниз. За окном моей кабины горизонт резко поднялся вверх. Самолет пикировал. Спидометр показывал скорость, превышающую 150 миль/ч. Я попытался воспользоваться горизонтальным рулем. Клавиша со знаком «плюс» слегка повернула руль вправо, и самолет начал вращаться, вызывая неприятные ощущения. Вернув руль в нейтральное положение, я стал наблюдать, как простирающийся подо мной ландшафт обманчиво медленно надвигается на меня. Настало время приступить к выполнению фигуры.

Пилоты входят в петли значительно реже, чем программисты в циклы. Интуиция подсказывала, что для маневра понадобится довольно большая скорость. Сначала вниз, затем вверх и потом вниз головой. Нажимать клавиши мне нравилось меньше, чем твердо поворачивать на себя ручку управления, но я тем не менее старательно нажимал на клавишу 2. Самолет с ревом вышел из пике в горизонтальный полет, а затем устремился вверх в голубую твердь небес. Я увидел, как линия горизонта ушла подо мной вниз, и стал ждать. У меня не было никакой уверенности в том, что фигура удастся. Все замерло в тишине. Может быть, самолет зависнет и рухнет на землю? Приборы не могли мне ничего подсказать, лишь о половине из них я имел какое-то представление, да и то смутное.

Внезапно на экране появился перевернутый ландшафт — сверху зеленая земля, а снизу голубое небо. Неужели маневр удался? Я был слишком разгорячен, чтобы ответить на свой вопрос. Полный мрачной решимости, я продолжал эксперимент. Теперь передо мной во все стороны раскинулась земля. Затем горизонт занял свое нормальное положение. Я попытался опустить нос самолета, чтобы предотвратить еще одну петлю. Горизонт покачнулся и замер на месте. Да, фигура удалась! Видимо, я принадлежу к числу тех одаренных людей, которые могут обучиться летному делу экспериментально, почти безо всякой предварительной подготовки. Впереди показался аэродром Мейгс-Филд. Как только совершу посадку, пойду приму душ.

Главная посадочная полоса аэродрома протянулась с севера на юг. Я находился к юго-востоку от аэродрома и решил лететь на запад, а порав-

WWW.SCIAM-MAGAZINE.NAROD.RU



Проекция четырехмерного гипердодекаэдра Э. Бекера

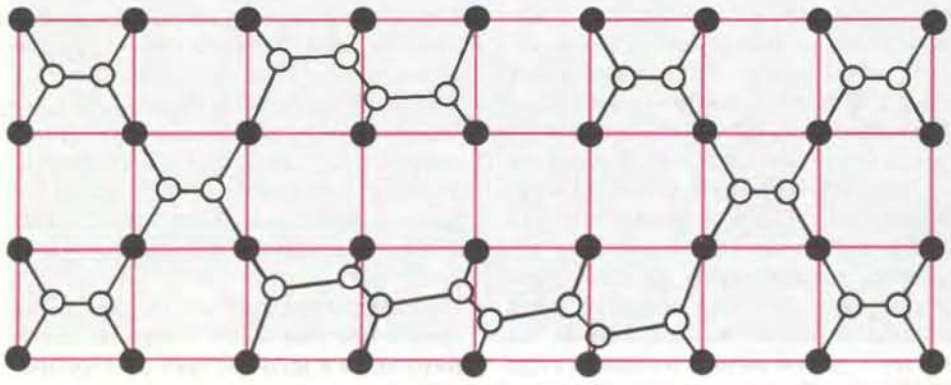
нявшись с посадочной полосой, повернуть на север. Реализовать этот план мне почему-то не удалось. Никак не удавалось зайти на посадочную полосу, и, кроме того, я приближался к аэродрому на слишком большой высоте. Все это было подобно кошмару, когда за одним несчастьем приходит другое. Я резко опустил нос самолета, чтобы побыстрее снизиться, стараясь в то же время дотянуть еще несколько метров в направлении на восток. Однако посадочная полоса ушла с экрана, и к тому же я потерял горизонт. Земля стала быстро приближаться, зазвучал сигнал тревоги. Черт побери! Я врезался в землю. Это стало очевидно, когда ветровое стекло кабины покрылось трещинами и на их фоне появилась крупная надпись «Авария». Погиб я или нет? Очень жаль, если база данных Чикаго лишится своего последнего обитателя.

Чтобы моделировать все эти приключения, FLIGHT SIMULATOR должна постоянно делать новые вычисления вида на ландшафт из кабины самолета. Рычаги управления самолетом можно сгруппировать по направлениям вращения вокруг трех осей (см. рисунок на с. 95). Так, элероны поворачивают самолет вокруг продольной оси, проходящей через носовую и хвостовую части самолета и называемой осью крена. Руль высоты вызывает поворот вокруг линии, параллельной крыльям, или оси тангажа (наклона). Руль направления поворачивает самолет вокруг вертикальной линии по отношению к пилоту, или оси рыскания. Если в какой-то момент положение самолета задано координатами x, y и z, то как программа определяет, где будет самолет мгновением позже?

Направление самолета и его скорость в данный момент образуют вектор, и по нему вычисляется новое положение машины. Программа определяет новое направление самолета (и, следовательно, новый угол зрения) из физических уравнений полета. На каждом цикле отображения программа задает новые начальные данные для этих уравнений, соответствующие текущему состоянию управляющих механизмов и условиям окружающей среды, таким, как давление и плотность воздуха, скорость ветра и температура. В уравнениях полета учтены физические параметры самолета, и к начальным данным уравнений применяются физические законы. Здесь работают такие силы, как вес, тяговое усилие двигателя, подъемная сила крыла и лобовое сопротивление воздуха. Учитываются также инерционные факторы, зависящие от массы и геометрической формы самолета. Решение уравнений дает новое направление полета.

Уравнения эти слишком сложны для того, чтобы микрокомпьютер мог решить их за приемлемое время, поэтому Артвик заменил их таблицами уже вычисленных значений. Одна из таблиц, например, содержит величины подъема самолета, соответствующие каждому дискретному значению угла атаки. Причем разность между двумя соседними значениями угла атаки составляет десятую долю градуса. Использование таблиц дает еще одно дополнительное преимущество: характер полета можно изменить, просто заменив таблицы, и новых формул выводить не надо.

Читатели, которых захватывают фигуры высшего пилотажа, могут воспользоваться готовой программой, имитирующей полет для любой распространенной марки домашнего компьютера. Артвик руководил созданием наиболее известной программы, распространяемой компанией Microsoft Corporation (Бельвю, шт. Вашингтон). Он сыграл также важную роль в разработке программы FLIGHT SIMULATOR II (Имитатор полета II). Обе программы могут вы-



Кратчайшее известное дерево Штейнера для массива точек 4 на 9

WWW.SCIAM-MAGAZINE.NARODRU

полняться компьютерами Apple II, IBM PC, Commodore 64 и Atari. Третья программа Артвика была создана недавно для персонального компьютера Macintosh.

После первых учебных полетов у читателей может возникнуть желание повторить захватывающие приключения, о которых повествуется в новой книге Ч. Гулика «Сорок грандиозных приключений в имитируемом полете». Каждое из описываемых в ней приключений начинается, когда читатель запускает любую из упомянутых выше программ имитации полета и вводит необходимые данные из книги Гулика. Эти данные включают в себя такие параметры, как широта, долгота, высота, положение управляющих механизмов, время дня и т. д.

Между прочим, вы можете оказаться в нескольких стах метрах от Манхаттенского моста и лететь по курсу, который приведет вас прямо под мост. Другая острая ситуация, «Мертвый номер у Сан-Клементе», заключается в том, что мотор самолета выходит из строя на высоте 5000 футов над островом Сан-Клементе у калифорнийского побережья Тихого океана. Сможете ли вы посадить самолет на планирующем полете? Третье приключение называется «Зона сумрака». Вы находитесь на травянистой взлетной полосе. Часы на приборной доске показывают три минуты первого ночи, но еще не совсем темно. Часть самолета, видимая из заднего бокового окна кабины, совершенно черная. Самолет медленно отрывается от земли, и после этого в окно уже ничего не видно. Никакой информации о ландшафте не дает и радар. Что бы вы стали делать в такой ситуации? Впрочем, лично я предпочитаю необитаемую базу данных.

«БОЙ В ПАМЯТИ» — тема моей первой статьи в этом журнале (см.: «В мире науки», 1984, № 7). Нынешним летом в последний уикенд августа на восточном побережье США Музей вычислительной техники в Бостоне провел первый международный турнир по этому виду соревнований компьютеров. Самые остроумные программы выводили из строя своих более слабых противников, уничтожая их ключевые команды. Победитель (человек, создавший победившую программу) получил звание чемпиона по бою в памяти и будет носить это звание в течение года.

Правила участия в чемпионате и требования, предъявляемые к программам, были разработаны Международной ассоциацией любителей игры «Бой в памяти», которая насчитывает около 150 членов. Возглавляет ассоциацию Марк Кларксон из Вичиты (шт Канзас).

Статья о программах, проходящих тесты на умственное развитие (см. «В мире науки», 1986, № 5), вдохновила многих читателей попробовать свои силы в мини-тесте. А некоторые написали даже свои собственные версии программы HI Q. П. Кук из Садбери (шт. Массачусетс), как и многие другие читатели, удивляется, как это удалось Марселю Феенстра, изобретателю программы HI Q, добиться результата в 160 очков. По словам самого Феенстра, эту идею подсказал ему его товарищ из Австралии К. Хардинг, набравший рекордную сумму баллов по тестам на умственное развитие. Хардинг предложил создать программу, которая могла бы подвергаться тестам, основанным на задачках по дополнению последовательностей, подобных тем, что содержатся в книге Г. Айзенка «Know Your Own I. Q». (Айзенк Г. Проверьте свои способности. — М.: Мир, 1972). В своей программе Феенстра принял систему оценки результатов, предложенную в книге Айзенка.

Статья о геометрических телах высших размерностей (см. «В мире науки» 1986, № 6) не вызвала у читателей никаких отклонений в психике. Все написавшие свою собственную версию программы HYPERCUBE сохранили душевное здоровье и физическую материальность. Правда, был выявлен поразительный случай ясновидения и предсказания будущего. Д. Эклейн из Рамни (шт. Нью-Гэмпшир) создал подобную программу за несколько месяцев до того, как я описал свою. Программа Эклейна дает все эффекты, производимые программой HYPERCUBE, но использует лишь четыре плоскости вращения вместо шести. Э. Холси из Сиэтла (шт. Вашингтон) не удовлетворился «мышиной возней» с программой HYPERCUBE. Он пошел дальше, к более сложным телам высоких размерностей, включая четырехмерный усеченный многогранник, четырехмерный симплекс и другие сложные объекты.

Наконец, следует отметить работу Э. Бекера из Риверхеда (шт. Нью-Йорк), который вплотную повстречался с гипердодекаэдром (см. рисунок вверху на с. 96). Обычный додекаэдр — это классическое геометрическое тело с 12 гранями. Гипердодекаэдр Бекера состоит из 120 обычных додекаэдров, построенных из 1200 линий. Фигура, изображенная на рисунке, была получена путем проектирования этого чудовищного объекта на трехмерное пространство с последующим смещением каждого из составляющих додекаэдров на небольшое расстояние в заданном направлении. Получается так, что каждая линия одновременно принадлежит трем додекаэдрам. Бекер пишет: «Плоские поверхности ограничены ребрами в их смещенном и несмещенном положениях. Соответственно я заменил каждую линию... чем-то вроде Y-образной рейки, в результате чего более понятной становится общая структура объекта, по крайней мере для моего трехмерного пространственного воображения».

В статье Мартина Гарднера (см. «В мире науки», 1986, № 8) перед читателями была поставлена задача найти дерево Штейнера, которое связывало бы матрицу из точек 4 на 9 и было бы короче 32,095 единицы. Пока никому не удалось улучшить результат, полученный Р. Грейемом и Ф. Чанг из фирмы Bell Laboratories (см. рисунок внизу на с. 96). Длина их дерева вычисляется по формуле $7[1 + \sqrt{3}] + [(([3(2 + \sqrt{3})] - 2)^2 + 1]^{1/2})/2] + [(([5(2 + \sqrt{3})] - 2)^2 + 1]^{1/2})/2]$, что приблизительно равно 32, 094656...

Издательство **МИР** *предлагает:*

Д. Хаузер, Дж. Хирт, Б. Хоукинс
ОПЕРАЦИОННАЯ СИСТЕМА MS-DOS:
Популярное руководство
Перевод с английского

MS-DOS — операционная система для персональных компьютеров, под управлением которой можно использовать тысячи прикладных программ. В книге подробно обсуждаются все команды MS-DOS, их формат, параметры, ключи. Рассматриваются возможности реконфигурации системы, стандартные средства отладки, редактирования, загрузки и печати. Кратко описаны утилиты MS-DOS.

Для пользователей ПК и студентов вузов.

1987, 14 л. Цена 1 р. 20 к.

ИЗДАТЕЛЬСТВО МИР